Available online at www.sciencedirect.com

SCIENCE @ DIRECT•

An International Journal
computers & mathematics
with applications

Computers and Mathematics with Applications 52 (2006) 211–224

www.elsevier.com/locate/camwa

Parallel Hybrid Algorithm for Global Optimization of Problems Occurring in MDS-Based Visualization

A. ŽILINSKAS* AND J. ŽILINSKAS
Institute of Mathematics and Informatics
Akademijos 4, Vilnius LT-08663, Lithuania
antanasz@ktl.mii.lt

Abstract—Multidimensional scaling is a widely used technique for visualization of multidimensional data. For the implementation of a multidimensional scaling technique a difficult global optimization problem should be solved. To attack such problems a hybrid global optimization method is developed combining evolutionary global search with local descent. A parallel version of the proposed method is implemented to enable solution of large scale problems in acceptable time. The results of the experimental investigation of the efficiency of the proposed method are presented.

Keywords—Multidimensional scaling, Metaheuristics, Global optimization, Parallel algorithms.

1. INTRODUCTION

Multidimensional scaling (MDS) is an exploratory technique for data analysis [1–3]. A set of n objects is considered assuming that pairwise dissimilarities of the objects are given by the matrix (δ_{ij}); depending on applications dissimilarities are determined by different empirical or computational methods, e.g., dissimilarities between several brands of cars can be evaluated comparing their technical characteristics, dissimilarities between tastes of several wines can be measured by subjective evaluations of a group of testers, etc. The m-dimensional image of the objects is a set of points in the embedding m-dimensional space $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$, $i = 1, \dots, n$. A set of points in the embedding space is sought whose interpoint distances fit the given dissimilarities δ_{ij}; we assume that $\delta_{ji} > 0$, and $\delta_{ij} = \delta_{ji}$. It is well known that under mild conditions there exist points in $(n-1)$-dimensional space whose pairwise distances are equal to dissimilarities; see, e.g., [2]. Therefore original data can be considered a set of points in a multidimensional vector space. We want to find an image of such a set in a low-dimensional embedding space. Since MDS is frequently used to visualize sets of points of multidimensional vector spaces, the

*Author to whom all correspondence should be addressed.

The authors acknowledge the support of Lithuanian State Science and Studies Foundation.
The work of the second author is supported by the NATO Reintegration Grant CBP.EAP.RIG.981300 and the HPC-Europa programme, funded under the European Commission's Research Infrastructures activity of the Structuring the European Research Area programme, contract number RII3-CT-2003-506079.

0898-1221/06/$ - see front matter Typeset by $\mathcal{AMS}$-TEX
doi:10.1016/j.camwa.2006.08.016

two-dimensional embedding space ($m = 2$) is of special interest. The implementation of an MDS method is reduced to minimization of a fitness criterion, e.g., the so-called *STRESS* function

$$S(X) = \sum_{i<j}^{n} w_{ij}(d_{ij}(X) - \delta_{ij})^2, \tag{1}$$

where $X = ((x_{11}, \dots, x_{n1}), \dots, (x_{1m}, \dots, x_{nm}))$; it is supposed that the weights are positive: $w_{ij} > 0$, $i, j = 1, \dots, n$; $d_{ij}(X)$ denotes the distance between the points $\mathbf{x}_i$ and $\mathbf{x}_j$. A norm in R^m should be chosen for calculation of distances. Most often a Minkowski distance is used

$$d_{ij}(X) = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^r \right)^{1/r}. \tag{2}$$

Formula (2) defines the Euclidean distances when $r = 2$, and the city block distances when $r = 1$. The points $\mathbf{x}_i$ defined by means of minimization of (1) but using different distances in the embedding space can be interpreted as different nonlinear projections of the objects from the original to the embedding space.

MDS is a difficult global optimization problem. Although *STRESS* is defined by an analytical formula, which seems rather simple, its minimization is difficult. The function normally has many local minima. The minimization problem is high-dimensional: number of variables is $N = n \times m$. Nondifferentiability of *STRESS* normally cannot be ignored. On the other hand, MDS-based visualization is useful in development of interactive global optimization methods [4].

The systematic application of global optimization in MDS has been initiated in [5], where majorization-based local search has been combined with the globalization of the descent by means of tunnelling. Further publications of different authors show great diversity of properties of MDS-related global optimization problems: depending on data, different algorithms have been shown more or less appropriate to solve MDS problems. For example, the advantages of genetic algorithms in MDS with nonstandard stress (fitness) criteria are discussed in [6]. The testing results in [7,8] prove that the hybrid algorithm combining evolutionary global search with efficient local descent is the most reliable though the most time-consuming method for MDS with Euclidean distances; the used test data well represent the typical visualization problems.

In the present paper we investigate efficiency/reliability of a combination of evolutionary global search with local descent in MDS applications to visualize multidimensional data. Sets of points in multidimensional vector spaces are visualized in the two-dimensional embedding space with Euclidean and city block norm. Parallelization of the algorithm is applied to cope with the large solution time of usual uniprocessor implementations.

2. BASIC VERSION OF HYBRID ALGORITHM

For the minimization of (1) a hybrid algorithm has been implemented combining a genetic algorithm (similar to that used in [9]) at upper level, and local minimization algorithm at lower level. The pseudocode of the algorithm is outlined below. The upper-level genetic algorithm ensures globality of search. Local descent at lower level ensures efficient search for local minima. From the point of view of evolutionary optimization the algorithm consists of the following 'genetic operators': random (with uniform distribution) selection of parents, two-point crossover, adaptation to environment (modelled by local minimization), and elitist survival. Interpreting the vector of variables in (1) as chromosome the crossover operator is defined by the following formula:

$$X = \text{arg\_min\_from}\,((\hat{x}_{11}, \dots, \hat{x}_{\xi_1 1}, \check{x}_{\xi_1+1\,1}, \dots, \check{x}_{\xi_2-1\,1}, \hat{x}_{\xi_2 1}, \dots, \hat{x}_{n1}),$$
$$(\hat{x}_{12}, \dots, \hat{x}_{\xi_1 2}, \check{x}_{\xi_1+1\,2}, \dots, \check{x}_{\xi_2-1\,2}, \hat{x}_{\xi_2 2}, \dots, \hat{x}_{n2})),$$

where X is the chromosome of the offspring; $\hat{X}$ and $\check{X}$ are chromosomes of the selected parents; ξ_1, ξ_2 are two integer random numbers with uniform distribution over $1, \ldots, n$; and it is supposed that the parent $\hat{X}$ is better fitted than the parent $\check{X}$ with respect to the value of *STRESS*. arg_min_from(Z) denotes an operator of calculation of the local minimizer of (1) from the starting point Z.

Outline of the pseudocode of the algorithm is presented below. The idea is to maintain a population of best (with respect to *STRESS* value) solutions whose crossover can generate better solutions. The size of population p is a parameter of the algorithm. An initial population is generated performing local search from p starting points that are best (with respect to the *STRESS* value) from a sample of N_{init} randomly generated points. The population evolves generating offsprings. Minimization terminates after predetermined computing time t_c.

The Structure of the Hybrid Algorithm with Parameters (p, N_{init}, t_c)

Generate the initial population:
  Generate N_{init} uniformly distributed random points.
  Perform search for local minima starting from the best p generated points.
  Form the initial population from the found local minimizers.
`while` t_c time has not passed
  Randomly with uniform distribution select two parents from a current population.
  Produce an offspring by means of crossover and local minimization.
  `If` the offspring is more fitted than the worst individual of the current population,
    `then` the offspring replaces the latter.

3. EXPERIMENTAL INVESTIGATION

Theoretical assessment of an optimization algorithm with respect of its efficiency in solution of MDS problems is difficult. For example, a local minimization subroutine could be evaluated according to the local convergence rate widely used in optimization theory. But this criterion represents only one of the efficiency aspects, and it is not sufficient to assess the general efficiency of global optimization. Therefore in this paper we investigate the efficiency of the proposed algorithm experimentally. To obtain results reproducible by the other researchers we use local minimization subroutines from an easily accessible library [10]. A Sun Fire E15k computer is used for experimental investigation.

3.1. Data Sets

Formally the quality of a result of multidimensional scaling can be assessed by the value of *STRESS* found by the minimization algorithm. However, in the visualization problems the heuristic acceptability of images is also very important. Several sets of multidimensional points corresponding to well-understood geometric objects are needed for the experimental investigation. We want to choose difficult test problems, i.e., difficult-to-visualize geometric objects. The data with desired properties correspond to the multidimensional objects equally extending in all dimensions of the original space, e.g., sets of vertices of multidimensional cubes and simplices. Dissimilarity between vertices is measured by the distance in the original vector space defined by its metric; we consider Euclidean and city block metrics. Global optimization problems of different difficulty can be constructed by defining dimensionality of the original spaces. Below we will sometimes use shorthand 'cube' and 'simplex' for sets of their vertices.

The number of vertices of a multidimensional cube is $n = 2^{\text{dim}}$, and the dimensionality of global minimization problem is $N = 2^{\text{dim}+1}$. The coordinates of the i^{th} vertex of a dim-dimensional cube are equal either to 0 or to 1, and they are defined by binary code of $i = 1, \ldots, n$. Vertices of multidimensional simplex can be defined by

$$v_{ij} = \begin{cases} 1, & \text{if } i = j+1, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, \dim +1, \quad j = 1, \ldots, \dim .$$

Dimensionality of this global minimization problem is $N = 2 \times (\dim +1)$.

Both types of geometric objects are special on symmetric location of vertices, and this feature is expected in the images. Besides this common feature, central location of the 'zero' vertex is characteristic for a simplex. Such a location of the image of the 'zero' vertex is expected in the visualized image of the set of simplex vertices. The vertices of a multidimensional cube compose clusters of 2^k points located equally distant from the center; the image is expected to highlight this feature.

All particular objects in the data sets are considered equally important; therefore all weights w_{ij} in (1) are set equal to one.

3.2. Impact of Metric

The MDS-based visualization quality depends on properties of *STRESS* and on precision of its minimization. The quadratic *STRESS* has been chosen as used most frequently in MDS literature. Norm in the embedding space can be chosen. We investigate two norms: Euclidean and city block. For the qualitative assessment we present best-known (with respect to the *STRESS* value) two-dimensional images of cubes and simplices of different dimensions. Besides qualitative assessment of informativeness of the images it is interesting to compare 'visualization errors' quantitatively. To exclude the impact of number of objects a relative error

$$f(X) = \sqrt{\frac{S(X)}{\sum_{i<j} \delta_{ij}^2}}$$

is used for comparison.

In Figures 1 and 2 the first column contains images obtained using Euclidean distances ($r = 2$) in original and embedding spaces, and the second column contains images using the city block metric ($r = 1$) in both spaces. The images of vertices are shown by circles. To make representations more visual, adjacent vertices are joined by lines. The darker lines show joints adjacent to two opposite vertices in the case of cubes and adjacent to the 'zero' vertex in the case of simplices.

Images of three-, four-, and five-dimensional cubes are shown in Figure 1. Vertices of multidimensional cubes tend to form a diamond-shaped structure when the city block metric is used. Let us note that points on the rhombus edges are of the same distance to the center in the city block metric; therefore vertices are visualized as expected. Moreover, images of vertices form clusters representing lower-dimensional cubes (sides and edges). In the case of the Euclidean metric, vertices of a cube tend to form clusters and fill a circle. In this case there is no uniformity in location of images of the vertices.

Images of eight-, twelve-, and sixteen-dimensional simplices are shown in Figure 2. The image of the 'zero' vertex is always located at the center of the structure. The images of the other vertices tend to form a rhombus- or diamond-shaped structure when the city block metric is used. Therefore they are located uniformly distant from the center. In the case of the Euclidean metric, the images of 'nonzero' vertices of low-dimensional simplices tend to form a circle. However when the dimensionality of simplex (and the number of vertices) increases, the next circle emerges, destroying uniformity of visualization of the 'nonzero' vertices.

The images of the multidimensional cubes and simplices obtained using the city block metric better highlight the structural properties of the original data than the images obtained using the

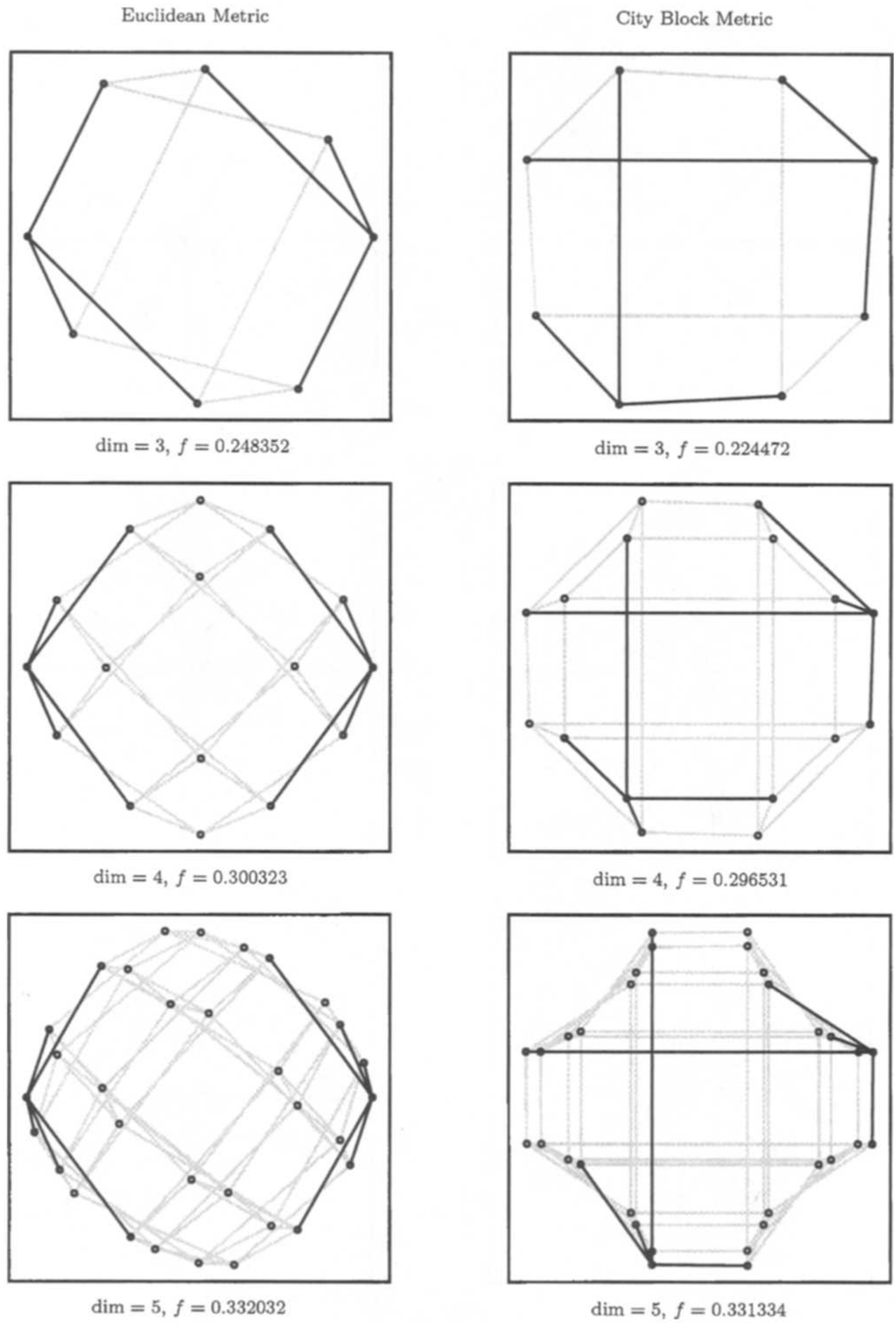

Figure 1. Images of cubes obtained using Euclidean and city block metrics.

Euclidean metric. The relative visualization errors f are shown above the figures. As it can be expected, errors increase with the dimensionality of objects, because it becomes more difficult to fit them into two-dimensional space. For the same dimensionality relative errors are smaller for images corresponding to the city block metric, however the difference decreases when the dimensionality of objects is increased.

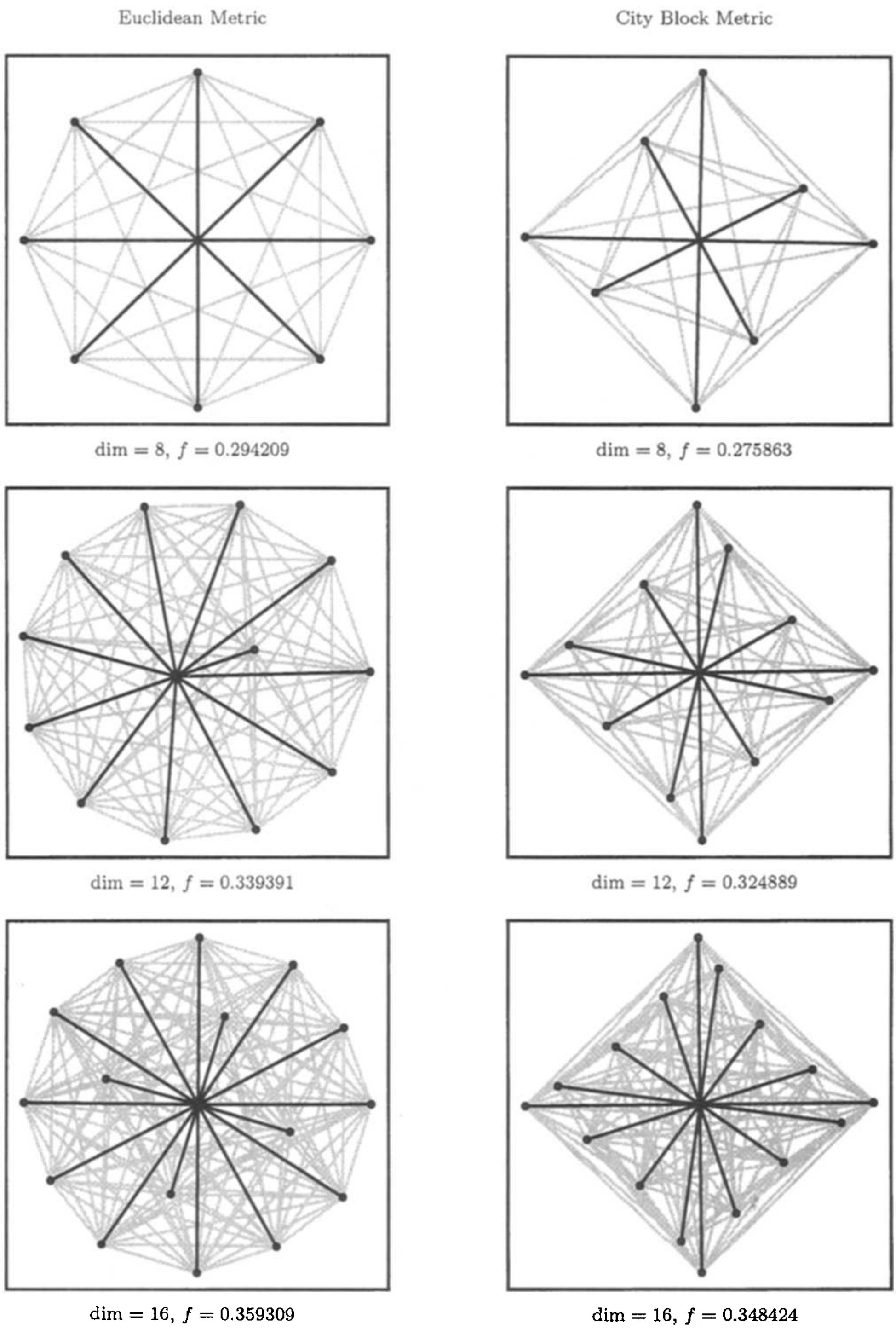

Figure 2. Images of simplices obtained using Euclidean and city block metrics.

3.3. Assessment of a Local Minimization Component

Although different local optimization techniques have been tried in MDS problems by many authors, there is no united opinion on efficiency of the tried techniques. For discussions on direct and surrogate function-based methods we refer to [11,12]. Moreover, the relative performance of the local minimization algorithm started from a random point not necessarily can be generalized to its relative performance when starting from rather close to local minimizers points generated by the genetic algorithm. Therefore we have performed an experimental investigation of the hybrid algorithm with two characteristic local minimization subroutines.

It is well known that *STRESS* with the Euclidean metric in embedding space is differentiable at the local minimizer [13]. Assuming smoothness of *STRESS* along a descent trajectory conjugate gradient method can be expected appropriate for minimization of *STRESS* with the Euclidean metric. The nondifferentiability of *STRESS* with the city block metric is analyzed in [14]; for the minimization of *STRESS* with the city block metric a direct search method (not using derivatives) is needed. For the experiments we have chosen two widely accessible algorithms: the conjugate gradient algorithm by Fletcher-Reeves-Polak-Ribiere, and direction set algorithm by Powell. The implementations of [10] were used. The conjugate gradient algorithm requests derivatives of the objective function; the approximation of derivatives by finite differences is applied, formally extending applicability of the algorithm to the case of the city block metric. The experiments have been performed to investigate difficulties caused by nondifferentiability of *STRESS* in the case of city block distances. In these experiments the parameters of the algorithm have been set to $p = 60$, $N_{\text{init}} = 6000$, $t_c = 10s$.

Let us start with the case of the Euclidean norm. Performance of the hybrid algorithm with the different local search methods at lower level can be assessed from the data presented in Table 1. Minimal, average, and maximal estimates of global minimum in 100 runs (f^*_{min}, f^*_{mean}, and f^*_{max}) are presented in the tables to show quality of the found solutions. The percentage of runs (perc.) when the estimate of the global minimum differs from f^*_{min} by less than 10^{-4} is presented in the table as a criterion of reliability of the algorithm. It can be seen from the table that the algorithm with Powell's local search performs better than with conjugate gradient local search. It is worth mentioning that during predefined ten seconds, more crossovers have been performed in the case when the conjugate gradient local search is used. The local search by the conjugate gradient takes less time than by Powell's method, but it seems that the conjugate gradient algorithm frequently terminates prematurely.

Performance of the hybrid method for MDS problems with the city block metric can be assessed from the estimates of the same criteria presented in Table 2. Again, the hybrid algorithm with Powell's local search performs better than with the conjugate gradient. The comparison of Tables 2 and 1 shows that the results for the problems with the city block metric are worse than the results for the problems with the Euclidean metric. We may conclude that although the local descent problem for the case of the city block metric is more complicated than for the case of the Euclidean metric, smoothness is not a sufficient criterion for assessment of complexity of optimization problems occurring in MDS.

One of the possible ways to improve the local search for MDS problems with the city block metric is to exploit special properties of *STRESS* implied by the considered metric. It is known that in this case *STRESS* is a piecewise (over simply defined polyhedrons) quadratic function of X. Therefore the local minimum over a polyhedron can be found by means of a quadratic programming method. However, the minimum point of a quadratic programming problem is not necessary a local minimizer of the initial problem (1); this is a case if a solution of a quadratic programming problem is on the border of the considered polyhedron. Therefore a combination of quadratic programming with unconstrained local search method (QP-ULS) seems reasonable: from the point found by the quadratic programming method to continue the local search ignoring the bounds of the quadratic programming problem [14]. Performance of the hybrid algorithm with QP-ULS in the problems with city block metric is shown in Table 3. Once again, QP-ULS with Powell's local search performs better than with conjugate gradient local search. The algorithm with QP-ULS with Powell's local search performs better than plain Powell's local search for cubes, but no significant change in performance for visualization of simplices is observed.

Table 1. Performance of the hybrid algorithm for MDS problems with the Euclidean metric.

	Conjugate Gradient Local Search				Powell's Local Search			
dim	$f^*_{\min}$	f^*_{mean}	$f^*_{\max}$	Perc.	$f^*_{\min}$	f^*_{mean}	$f^*_{\max}$	Perc.
				Cubes				
3	0.2439	0.2439	0.2439	100	0.2439	0.2439	0.2439	100
4	0.3003	0.3003	0.3004	99	0.3003	0.3003	0.3003	100
5	0.3343	0.3379	0.3421	1	0.3320	0.3321	0.3330	74
6	0.3746	0.4286	0.4470	1	0.3505	0.3509	0.3532	41
				Simplices				
5	0.2122	0.2122	0.2122	100	0.2122	0.2122	0.2122	100
6	0.2482	0.2482	0.2482	100	0.2482	0.2482	0.2482	100
7	0.2744	0.2744	0.2744	100	0.2744	0.2744	0.2744	100
8	0.2942	0.2942	0.2942	100	0.2942	0.2942	0.2942	100
9	0.3097	0.3097	0.3097	100	0.3097	0.3097	0.3097	100
10	0.3221	0.3221	0.3221	100	0.3221	0.3221	0.3221	100
11	0.3317	0.3317	0.3317	100	0.3317	0.3317	0.3317	100
12	0.3394	0.3394	0.3395	95	0.3394	0.3394	0.3394	100
13	0.3457	0.3457	0.3458	94	0.3457	0.3457	0.3457	100
14	0.3509	0.3509	0.3511	68	0.3509	0.3509	0.3509	100
15	0.3554	0.3555	0.3556	44	0.3554	0.3554	0.3554	100
16	0.3593	0.3595	0.3598	25	0.3593	0.3593	0.3593	100
17	0.3628	0.3630	0.3633	15	0.3628	0.3628	0.3628	100
18	0.3660	0.3662	0.3664	7	0.3660	0.3660	0.3660	100
19	0.3688	0.3691	0.3693	7	0.3687	0.3687	0.3687	100
20	0.3714	0.3716	0.3718	3	0.3713	0.3713	0.3713	100

4. PARALLEL VERSION OF HYBRID ALGORITHM

Experiments in the previous section have shown that visualization of a six-dimensional cube is already a difficult 128-dimensional problem (because of $n = 64$ objects in the original space). Larger problems would be even more difficult. When the computing power of the usual computers is not sufficient to solve global optimization problems of so many variables, the high performance parallel computers may be helpful. Therefore a parallel version of the hybrid algorithm for large-scale MDS has been developed.

A parallel version of a genetic algorithm with multiple populations [15] has been developed. Communications between processors have been kept to minimum to enable implementation of the algorithm on clusters of personal computers. Each processor runs the same genetic algorithm with different sequences of random numbers. The results of different processors are collected when the search is finished after a predefined time.

To make parallel implementation as portable as possible the general message-passing paradigm of parallel programming has been chosen. A standardized message-passing communication protocol MPI [16] is used for communication between parallel processors. A Sun Fire E15k computer is used for experimental investigation.

In the experiments whose results are described below the hybrid algorithm with Powell's method as a local component has been used for the problems with the Euclidean metric; QP-ULS

Table 2. Performance of the hybrid algorithm for MDS problems with the city block metric.

	Conjugate Gradient Local Search				Powell's Local Search			
dim	f^*_{min}	f^*_{mean}	f^*_{max}	Perc.	f^*_{min}	f^*_{mean}	f^*_{max}	Perc.
				Cubes				
3	0.2245	0.2245	0.2245	100	0.2245	0.2245	0.2245	100
4	0.2965	0.2967	0.2999	83	0.2966	0.2968	0.2974	19
5	0.3332	0.3380	0.3494	1	0.3315	0.3320	0.3350	5
6	0.4163	0.4788	0.5157	1	0.3516	0.3561	0.3784	5
				Simplices				
5	0.1869	0.1869	0.1869	100	0.1869	0.1869	0.1869	100
6	0.2247	0.2247	0.2247	100	0.2247	0.2247	0.2247	100
7	0.2569	0.2569	0.2569	100	0.2569	0.2569	0.2569	100
8	0.2759	0.2759	0.2759	100	0.2759	0.2759	0.2759	100
9	0.2936	0.2936	0.2936	100	0.2936	0.2936	0.2936	100
10	0.3058	0.3058	0.3058	100	0.3058	0.3058	0.3058	100
11	0.3167	0.3167	0.3177	97	0.3167	0.3167	0.3167	100
12	0.3249	0.3250	0.3259	84	0.3249	0.3249	0.3249	100
13	0.3325	0.3327	0.3338	69	0.3325	0.3325	0.3328	99
14	0.3384	0.3388	0.3400	45	0.3384	0.3385	0.3389	96
15	0.3439	0.3442	0.3453	46	0.3439	0.3441	0.3445	53
16	0.3484	0.3488	0.3497	34	0.3484	0.3487	0.3493	34
17	0.3526	0.3531	0.3543	14	0.3526	0.3530	0.3536	7
18	0.3562	0.3567	0.3578	10	0.3562	0.3566	0.3573	2
19	0.3595	0.3602	0.3616	3	0.3595	0.3600	0.3604	1
20	0.3623	0.3634	0.3646	6	0.3624	0.3629	0.3634	3

including Powell's subroutine has been used as a local minimization subroutine for the problems with the city block metric.

4.1. Improvement of Performance

For the assessment of the performance of the parallel version (eight processors) of the hybrid algorithm, minimization results of geometric problems are presented in Table 4; the more difficult problems with the city block metric have been used. Performance improvement is significant for all considered problems comparing with the performance on a single processor shown in Table 3. Reliability of visualization of a five-dimensional cube is increased four times, and reliability of visualization of 16-dimensional simplex is increased seven times. Parallelization has increased dimensionality of reliably visualized simplices from 12 to 14.

Figure 3 shows how perc. (the percentage of runs finding the best known solution) depends on the number of processors used. The curves represent the problem of different dimensionality shown in the previously discussed tables. The curve in the figure is located higher when the corresponding problem is solved more reliably. Naturally, higher-located curves represent problems with smaller dimensionality. Generally speaking, performance of the parallel algorithm increases when the number of processors used is increasing. The reliability of the solution of the largest problems (especially visualization of simplices) is not sufficient in all cases, i.e., using up to eight processors. For such problems either the number of processors or solution time should be noticeably increased.

Table 3. Performance of the hybrid algorithm with local search strategy based on quadratic programming for MDS problems with city block metric.

	Conjugate Gradient Local Search				Powell's Local Search			
dim	$f^*_{\min}$	f^*_{mean}	$f^*_{\max}$	Perc.	$f^*_{\min}$	f^*_{mean}	$f^*_{\max}$	Perc.
				Cubes				
3	0.2245	0.2245	0.2245	100	0.2245	0.2245	0.2245	100
4	0.2965	0.2967	0.2978	76	0.2965	0.2965	0.2969	96
5	0.3337	0.3494	0.3651	1	0.3313	0.3317	0.3354	14
6	0.3650	0.3904	0.4068	1	0.3514	0.3577	0.3784	1
				Simplices				
5	0.1869	0.1869	0.1869	100	0.1869	0.1869	0.1869	100
6	0.2247	0.2247	0.2247	100	0.2247	0.2247	0.2247	100
7	0.2569	0.2569	0.2569	100	0.2569	0.2569	0.2569	100
8	0.2759	0.2759	0.2759	100	0.2759	0.2759	0.2759	100
9	0.2936	0.2936	0.2936	100	0.2936	0.2936	0.2936	100
10	0.3058	0.3058	0.3058	100	0.3058	0.3058	0.3058	100
11	0.3167	0.3168	0.3177	94	0.3167	0.3167	0.3167	100
12	0.3249	0.3250	0.3259	89	0.3249	0.3249	0.3249	100
13	0.3325	0.3327	0.3337	56	0.3325	0.3325	0.3330	93
14	0.3384	0.3388	0.3402	48	0.3384	0.3386	0.3391	70
15	0.3439	0.3443	0.3455	34	0.3439	0.3443	0.3448	25
16	0.3484	0.3490	0.3506	19	0.3484	0.3490	0.3497	8
17	0.3526	0.3532	0.3539	9	0.3526	0.3532	0.3538	3
18	0.3562	0.3568	0.3582	7	0.3562	0.3568	0.3575	2
19	0.3595	0.3602	0.3618	1	0.3597	0.3602	0.3607	4
20	0.3626	0.3633	0.3643	7	0.3625	0.3630	0.3636	4

4.2. Applications to Large Scale Problems

The developed parallel hybrid algorithm with the parameter $t_c = 2h$ can be applied to visualize large scale problems. The performance parameters are estimated from 10 runs performed for each problem.

First we consider geometric data of higher dimensionality than in the previous section. The images of a six-dimensional cube and 63-dimensional simplex are shown in Figure 4, and the performance parameters in Table 5.

Additionally to multidimensional geometric objects two data sets of the problems widely discussed in MDS-related literature have been chosen for the experiment. The structure of the considered real-world data, although not an intrinsic geometric property of the set of points, is well researched by numerous investigators using various methods.

The first set is Iris data [17]. The data set consists of 150 instances (50 in each of three classes: Iris Setosa, Iris Versicolour, Iris Virginica). Four numeric attributes (sepal length, sepal width, petal length, petal width) define points in the four-dimensional original space. The parameters of the data set are: dim $= 4$, $n = 150$, $N = 300$. Iris data has been analyzed by numerous authors using different methods, and it is well known that the set of points consists of two contiguous clusters and one well-separated cluster. Images of Iris data obtained by means of MDS are expected to highlight this structure.

The second data set is Morse code confusion data originally presented by a proximity matrix; see, e.g., [1]. Dissimilarity can be defined via proximity in different ways. We have used a

Table 4. Performance of parallel version (eight processors) of the hybrid algorithm in MDS problems with the city block metric; local search strategy combines quadratic programming with Powell's method.

dim	$f^*_{\min}$	f^*_{mean}	$f^*_{\max}$	Perc.
		Cubes		
3	0.2245	0.2245	0.2245	100
4	0.2965	0.2965	0.2965	100
5	0.3313	0.3314	0.3316	55
6	0.3513	0.3516	0.3522	7
		Simplices		
5	0.1869	0.1869	0.1869	100
6	0.2247	0.2247	0.2247	100
7	0.2569	0.2569	0.2569	100
8	0.2759	0.2759	0.2759	100
9	0.2936	0.2936	0.2936	100
10	0.3058	0.3058	0.3058	100
11	0.3167	0.3167	0.3167	100
12	0.3249	0.3249	0.3249	100
13	0.3325	0.3325	0.3325	100
14	0.3384	0.3384	0.3384	100
15	0.3439	0.3439	0.3443	94
16	0.3484	0.3486	0.3490	56
17	0.3526	0.3529	0.3531	17
18	0.3562	0.3565	0.3568	5
19	0.3595	0.3599	0.3602	2
20	0.3623	0.3627	0.3631	2

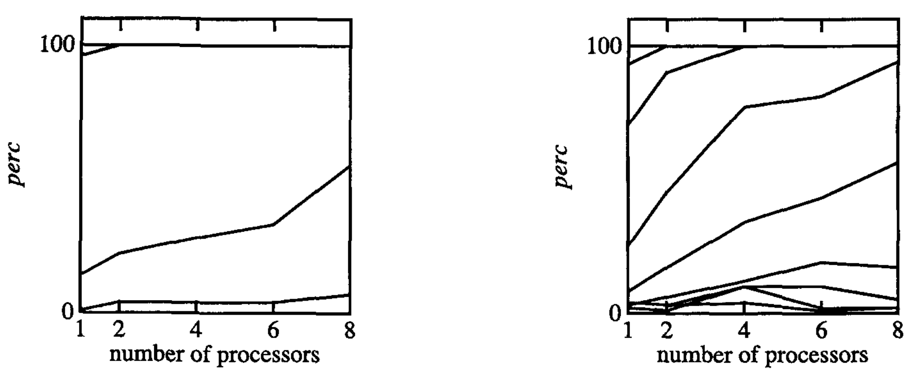

Figure 3. Performance improvement using parallel algorithm; left graphs for cubes, right graphs for simplices.

dissimilarity matrix calculated from the proximity matrix according to the formula of [18]. The considered set (Morse codes of Latin letters and of numerals) consists of $n = 36$ objects. The dimensionality of the global optimization problem is $N = 72$.

The images of Iris data are shown in Figure 5. Different classes of Iris data are denoted by the letters t (Iris Setosa), l (Iris Versicolour), and g (Iris Virginica). Values of the *STRESS* function are shown above the figures. The known structure of the data is very visible in both pictures.

Table 5. Performance of the parallel version of the hybrid algorithm (eight processors) in visualization of large data sets.

	Euclidean Metric				City Block Metric			
N	f^*_{min}	f^*_{mean}	f^*_{max}	Perc.	f^*_{min}	f^*_{mean}	f^*_{max}	Perc.
				6-Dimensional Cube				
128	0.3505	0.3505	0.3505	100	0.3513	0.3513	0.3513	100
				63-Dimensional Simplex				
128	0.4051	0.4051	0.4051	100	0.3998	0.3998	0.3998	100
				Iris Data				
300	0.03273	0.03273	0.03273	100	0.04710	0.04710	0.04710	100
	109.418	109.421	109.427		637.091	637.103	637.135	
				Morse Code Confusion Data				
72	0.3001	0.3001	0.3001	100	0.2944	0.2944	0.2944	100
	159.005	159.005	159.005		153.001	153.001	153.001	

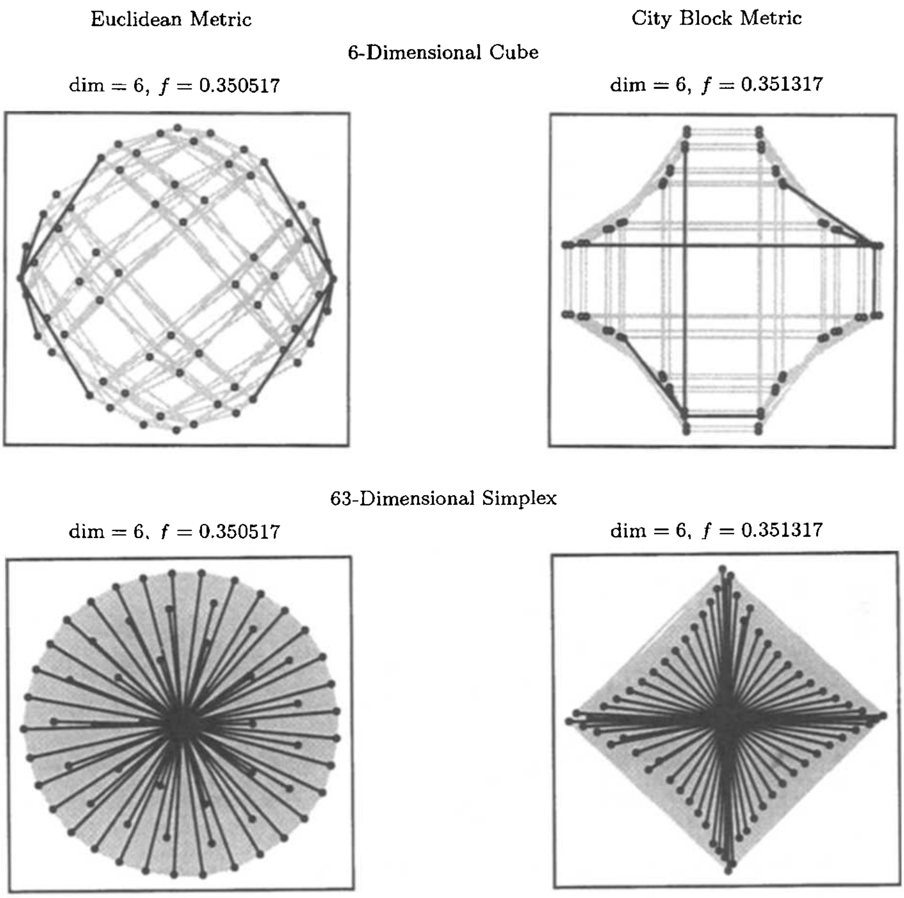

Figure 4. Images of large geometric objects visualized using MDS.

The images of Morse code confusion data are shown in Figure 5. Objects of different codes are represented by corresponding letters and numerals. Images of Morse code confusion data remind of images of a cube. Values of the *STRESS* function are shown above the figures. In the case of the city block metric a better value of *STRESS* than 153.24 (the record value of previously published results given in [18]) has been found.

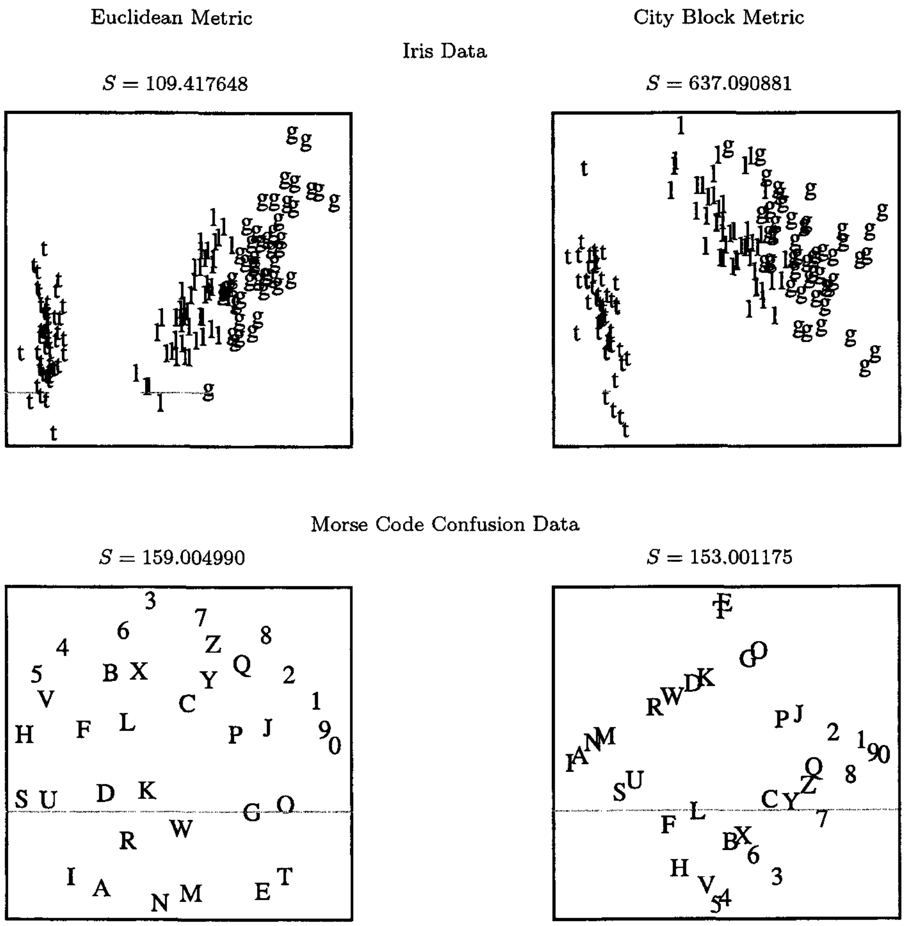

Figure 5. Images of large practical problems visualized using MDS.

Performance of the parallel version of the hybrid algorithm (eight processors) in visualization of large scale real world problems is shown in Table 5. As well as relative errors, the values of the *STRESS* function are presented for Iris and Morse problems.

5. CONCLUSIONS

The implementation of MDS-based visualization methods can be reduced to global minimization of *STRESS*. The properties of the minimization problem essentially depend on norms used in original and embedding spaces, e.g., the minimization problem in the case of the city block norm is more difficult than in the case of the Euclidean norm. However, the images corresponding to the former case better highlight properties of the geometric originals than the images corresponding to the latter case. The relative visualization errors also are less when the city block metric is used than when the Euclidean norm is used.

A hybrid algorithm combining genetic global search with local minimization is proposed, and suitable local minimization subroutines are experimentally selected for the cases of city block and Euclidean norms. In both cases the minimization of difficult optimization problems up to 30 variables corresponding to visualization of geometric data are reliably solved. A parallel version of the proposed hybrid algorithm is developed applicable to a reliable solution of global optimization problems up to 300 variables corresponding to visualization of real-world data.

~~A record-breaking *STRESS* value has been found for the Morse code confusion problem with~~ the city block metric.

REFERENCES

1. I. Borg and P. Groenen, *Modern Multidimensional Scaling*, Springer, New York, (1997).
2. T. Cox and M. Cox, *Multidimensional Scaling*, Chapman and Hall/CRC, Boca Raton, FL, (2001).
3. J. De Leeuw and W. Heiser, Theory of multidimensional scaling, Volume 2, In *Handbook of Statistics*, (Edited by P.R. Krishnaiah), pp. 285–316, North-Holland, Amsterdam, (1982).
4. A. Törn and A. Žilinskas, *Global Optimization, Lecture Notes in Computer Science 350*, pp. 1–250, Springer, Berlin, (1989).
5. P. Groenen, *The Majorization Approach to Multidimensional Scaling*, DSWO, Amsterdam, (1993).
6. J.E. Everett, Algorithms for multidimensional scaling, In *The Practical Handbook of Genetic Algorithms*, (Edited by L. Chambers), pp. 203–233, Chapman and Hall/CRC, Boca Raton, FL, (2001).
7. P. Groenen, R. Mathar and J. Trejos, Global optimization methods for MDS applied to mobile communications, In *Data Analysis: Scientific Models and Practical Applications*, (Edited by W. Gaul, O. Opitz and M. Schander), pp. 459–475, Springer, (2000).
8. R. Mathar, A hybrid global optimization algorithm for multidimensional scaling, In *Classification and Knowledge Organization*, (Edited by R. Klar and O. Opitz), pp. 63–71, Springer, Berlin, (1996).
9. R. Mathar and A. Žilinskas, On global optimization in two-dimensional scaling, *Acta Applicandae Mathematicae* **33**, 109–118, (1993).
10. W. Press *et al.*, *Numerical Recipes in C++*, Cambridge University Press, Cambridge, (2002).
11. A. Kearsley, R. Tapia and M. Trosset, The solution of the metric STRESS and SSTRESS problems in multidimensional scaling using Newton's method, *Computational Statistics* **13**, 369–396, (1998).
12. K. Lange, D. Hunter and I. Yang, Optimization transfer using surrogate objective functions (with discussion), *Journal of Computational and Graphical Statistics* **9** (1), 1–59, (2000).
13. J. De Leeuw, Differentiability of Kruskal's stress at a local minimum, *Psychometrika* **149**, 111–113, (1984).
14. A. Žilinskas and J. Žilinskas, Two level minimization in multidimensional scaling, *Journal of Global Optimization*, (submitted).
15. E. Cantú-Paz, *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic, (2000).
16. Message Passing Interface Forum, MPI: A message-passing interface standard (version 1.1), Technical Report, (1995).
17. R.A. Fisher, The use of multiple measurements in taxonomy problems, *Annals of Eugenics* **7**, 179–188, (1936).
18. M.J. Brusco, A simulated annealing heuristics for unidimensional and multidimensional (city block) scaling of symmetric proximity matrices, *Journal of Classification* **18**, 3–33, (2001).